测试需求分析

1. **测试需求是什么**
2. 解决测什么，一般来自规格说明书中的原始需求；测试需求需要覆盖已定义的业务流程，以及功能和非功能方面的需求
3. 为什么要软件测试需求

——只有明确需求，才能知道怎么去测，时间，人力，环境要求

**测试点思路步骤如下：**

1. **、**正常功能：是否可以提交
2. **单个功能（输入）项验证（正常+异常）**

规则：按顺序从上而下、对每一个输入项进行验证

* 数据长度（或范围）、数据类型验证、必填项验证、重复
* 限制约束验证
* 隐性需求：充分熟悉产品业务、挖掘隐性需求——不强求

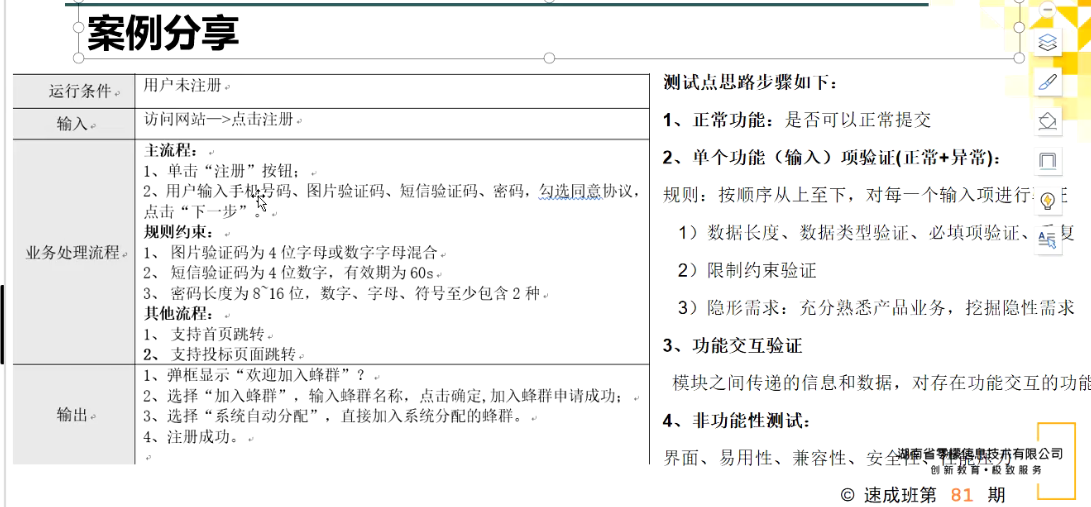
1. **功能交互验证——没有功能关联的功能，就不要考虑了**

板块之间传递的信息和数据，对存在功能交互的功能项，比如添加购物车，数量1变成2时，要看下数据是否同步（以上3点都属于功能测试）

1. 、非功能性测试：

界面、易用性、兼容性、安全性、性能压力测试

初中级测试工程师考虑安全方面——敏感信息，是否加密；再考虑性能负载，

案例：

1. 笔试面试题整理
2. 、遇到隐性需求怎么办？——遇到需求上的问题，一定要找产品经理确认
3. 、给你一个带有logo的水杯（面试的时候发散性思维，即对生活中的某个物品进行测试，比如桌子、门、伞），你会如何去测试？

思路：功能和非功能

功能测试：能否正常装水，能装多少水，能不能装其他方面液体/固体，

非功能性：

界面测试：logo是否正确显示，杯子是否光滑明亮，logo的位置是否显示在中间；

易用性测试：杯子大小，即是否容易被握住，是否有杯垫。。。

兼容性测试：本身材质考虑，放到微波炉是否会炸，冷冻柜里会不会炸，

安全测试：材质是否有毒，或加热后是否有毒，杯子是否容易划伤

性能测试：从多少米的高空掉落会碎，用多大的力去握住会碎

1. 、你会如何去测试朋友圈、购物车等熟知的软件产品。（支付，优惠券，二维码）

**用例方法设计之四大金刚**

1. 等价类划分方法

——黑盒测试，指某个输入域的子集合，在该子集合中，所有的输入数据对于揭露软件中的错误都是等效的

* 有效等价类：输入的数据对于软件而言，符合需求的，是合理的
* 无效等价类：数据对于软件而言是不符合需求、不合理的，

举例：微信红包

* 按数据范围划分：有效的：0.01--200，（1） 不超过2位小数的值

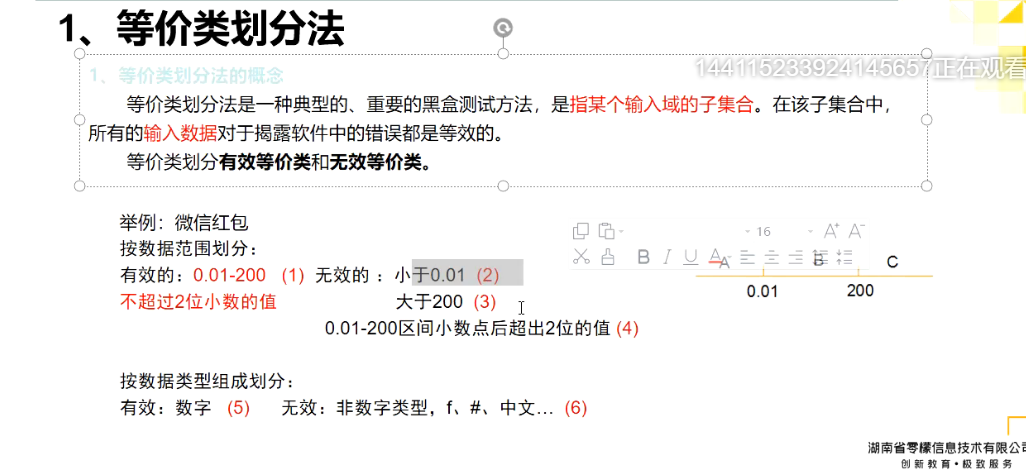
无效的：小于0.01，（2）大于200（3）

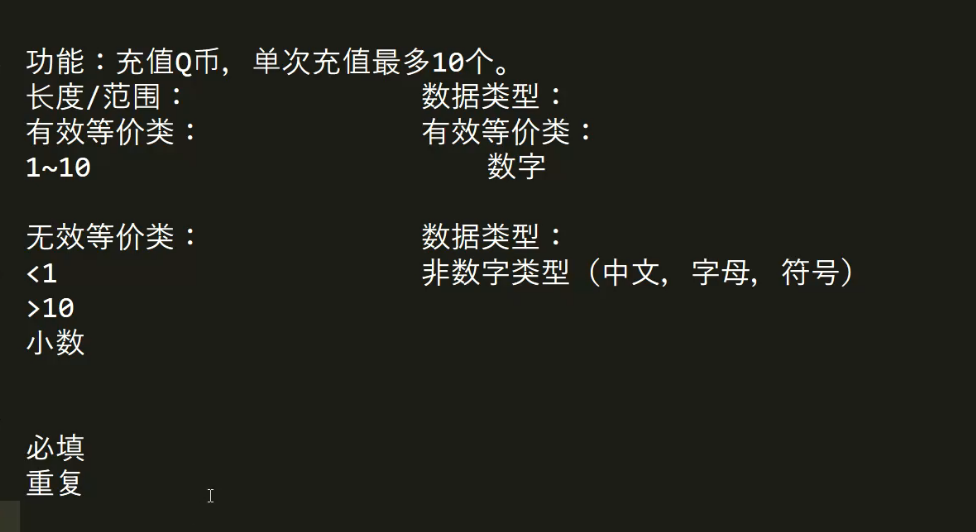
0.01--200区间小数点后超出2位的值（4）

* 按数据类型划分：

有效：数字 （5） 无效：非数字类型，f，#，中午。。。（6）

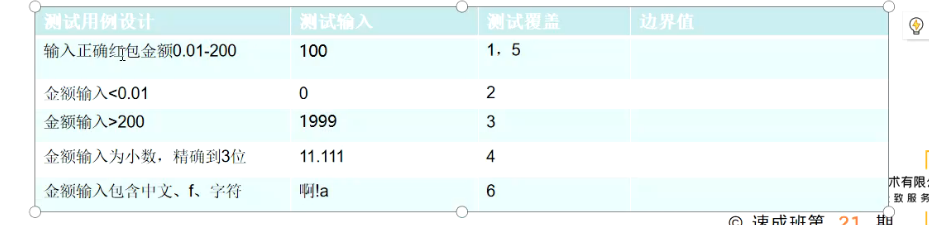
假设，字母（7）





1. 等价类划分法用例设计原则

* 划分有效及无效，为每一个等价类规定一个唯一的 编号
* 设计一个新的测试用例数据，使其尽可能多的覆盖尚未被覆盖的有效等价类，重复这一步，直到所有有效等价类都被覆盖为止
* 设计一个新的测试用例数据，使其仅覆盖一个尚未被覆盖的无效等价类，重复，直到无效的都被覆盖为止（为何无效不能组合：不清楚哪个环节出问题）
* 基本用例设计



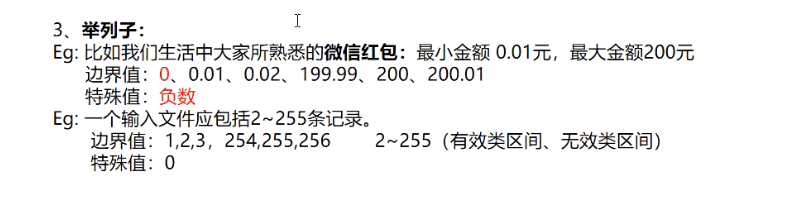
1. 边界值分析法

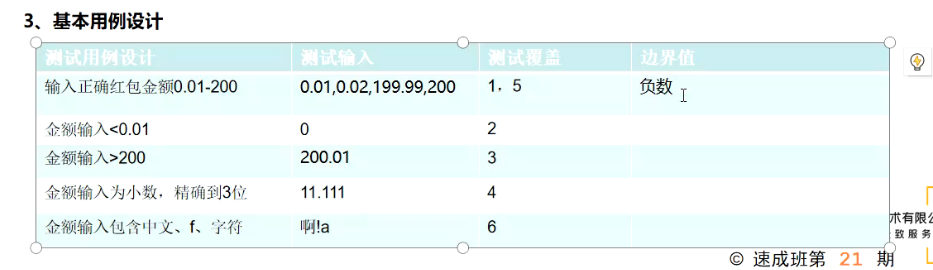
——对等价类的补充，边界值一般是从等价类的边缘值去寻找，其基本思想是：正好等于、刚刚大于、刚刚小于的边界值作为测试数据。

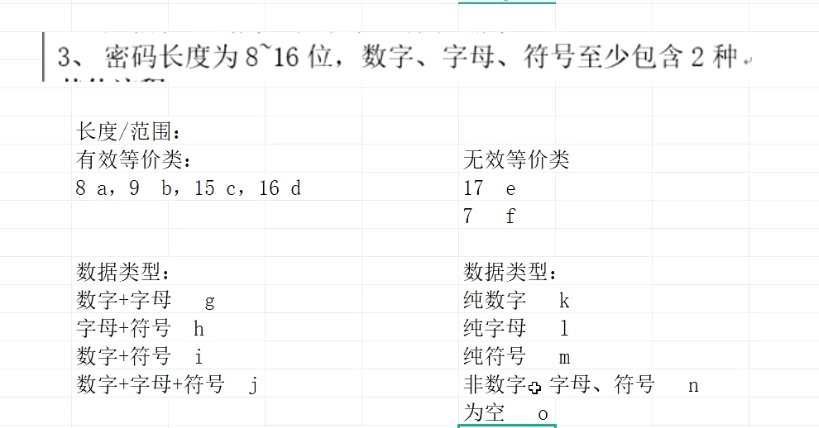
* 注意：有两个特殊值，负数：只运用在输入框是数值型的地方；0：金额或者空值

作用：错误大多发生在输入或者输出的边界的，为方便查找更多的错误

比如：微信，0.01--200





* 案例： 假如abc组合ghij，剩下的j随便组合，类似a/ba\/c+j ； 加入超过16，等价类有效值内选择匹配，比如10。

1. **等价类和边界值常用的场景**
2. 、输入规定的取值范围或值的个数情况（类似最小小于x小于最大，最小小于x，最大大于x，比如用户名长度和红包金额数值输入范围）
3. 在输入条件是true和false两种状态下的情况：比如勾选，开关设置
4. 在下拉列表包含多个选项的情况，比如城市下拉选项（第一个、最后一个、中间一个
5. 在规定了输入数据必须遵守的规则情况下，可确立一个有效等价类，（符合规则）和若干个无效等价类（从不同角度违反规则）

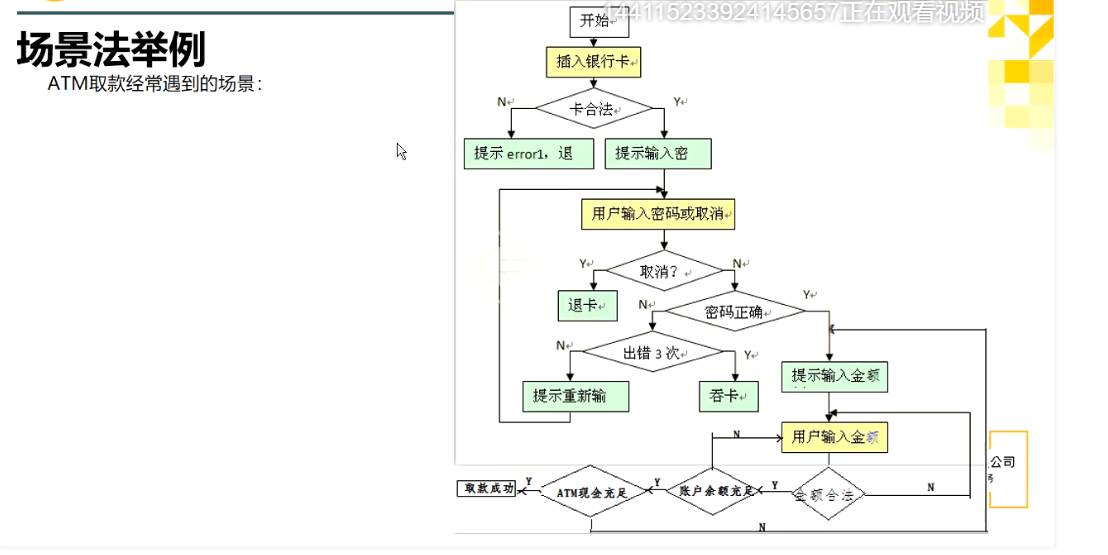
以上除了等价类之外同时还会涉及边界值分析，边界值还包括：

1. 报表数据的第一、中间、最后一行
2. 屏幕光标在最左上、最右下位置
3. **场景法**
4. 、——通过场景描述的业务流程（业务逻辑）、也包括代码实现逻辑，设计用例来遍历场景（路径），验证软件系统正确性
5. 如何使用场景：

画出流程图：

1. 矩形：表述步骤/数据
2. 菱形：判断Y/N，只有两个出口，
3. 箭头：流向

注意：场景法重点是测试流程，所以每个流程用一个用例验证即可，流程没问题不能说明系统功能没问题了，还需要很对单步的功能进行测试，只有单个功能点和流程测试，才是充分的测试



1. 错误推测法（反推法）

——经验和直觉、知识，推测可能存在的问题，再看哪些是测试过的（用等价类和场景法），保留没有考虑过的，进行补充用例设计。

